

WO0046753

Publication Title:

Method and apparatus for 3D model creation based on 2D images

Abstract:

A method in accordance with the invention comprises the step of providing a) pixel array and a computer model of a geometric surface, and b) providing a second array (a Z array) containing information concerning the 3D characteristics of the structure depicted in the pixel array. The Z array is used to modify the location of the geometric surface. The pixel array and modified geometric surface are then used to render an image that can be displayed on a CRT or other output device. By modifying the geometric surface model in this way, it is possible to create more realistic images

Data supplied from the esp@cenet database - <http://ep.espacenet.com>



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06T 11/40		A1	(11) International Publication Number: WO 00/46753 (43) International Publication Date: 10 August 2000 (10.08.00)
<p>(21) International Application Number: PCT/US00/02786</p> <p>(22) International Filing Date: 2 February 2000 (02.02.00)</p> <p>(30) Priority Data: 60/118,508 3 February 1999 (03.02.99) US 09/361,470 27 July 1999 (27.07.99) US </p> <p>(71) Applicant: ISURFTV [US/US]; Suite 307, 1975 El Camino Real West, Mountain View, CA 94040 (US).</p> <p>(72) Inventors: KAMEN, Yakov; 19334 Greenwood Drive, Cupertino, CA 95014 (US). SHIRMAN, Leon; 870 Seminole Way, Redwood City, CA 94062 (US).</p> <p>(74) Agent: LEEDS, Kenneth; P.O. Box 2819, Sunnyvale, CA 94087-0819 (US).</p>		<p>(81) Designated States: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).</p> <p>Published</p> <p><i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>	
<p>(54) Title: METHOD AND APPARATUS FOR 3D MODEL CREATION BASED ON 2D IMAGES</p> <p>(57) Abstract</p> <p>A method in accordance with the invention comprises the step of providing a) a pixel array (100) and a computer model of a geometric surface (102), and b) providing a second array (a Z array) containing information concerning the 3D characteristics of the structure depicted in the pixel array (100). The Z array (104) is used to modify the location of the geometric surface (102). The pixel array (100) and modified geometric surface (102) are then used to render an image that can be displayed on a CRT or other output device. By modifying the geometric surface model (102) in this way, it is possible to create more realistic images.</p>			
<pre> graph TD A["PROVIDE Z ARRAY 104, GEOMETRIC SURFACES 102 AND PIXEL ARRAY 100"] --> B["BIND Z ARRAY 104 TO GEOMETRIC SURFACES 102"] B --> C["MODIFY GEOMETRIC SURFACES 102 USING Z ARRAY INFORMATION"] C --> D["BIND PIXEL ARRAY 100 TO MODIFIED GEOMETRIC SURFACES 102"] D --> E["USE GRAPHICS PIPELINE TO RENDER 3D IMAGE"] E --> F["PROVIDE PIXEL ARRAY 116 IN FRAME BUFFER"] </pre>			

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

1

2

3

4

5

6 METHOD AND APPARATUS FOR 3D MODEL
7 CREATION BASED ON 2D IMAGES

8

9

10

11

12 This patent claims priority based on U.S. Patent Application Serial No.

13 60/118,508, filed 2/3/99, and Application Serial No. 09/361,470, filed 7/27/99, each

14 incorporated herein by reference.

15 BACKGROUND OF THE INVENTION

16 This invention pertains to 3D object construction. In particular, this invention

17 pertains to integrated texture mapping, filtering and image geometry displacement.

18 It is known in the art to computer generate 3D objects, texture them, and project
19 them as a sequence of images on a screen. One way of doing this is with a conventional
20 3D graphics pipeline. Briefly, a conventional 3D graphics pipeline creates an image by
21 performing the following tasks:

22 1. A computer model of a geometric surface is created or provided. The computer
23 model can be an array of polygons, described in the computer model in terms of
24 the x, y and z coordinates of its vertices. The polygons are joined together at their
25 edges to form a 3D surface. Alternatively, the computer model can be a set of
26 geometric surfaces that are defined in other ways, e.g. "implicitly defined," using
27 mathematical equations.

28 2. A pixel array (or arrays) containing one or more images to be applied to the
29 polygons (or implicitly defined geometric surfaces) as textures is provided. We

1 will refer to this type of pixel array as a "texel" array. We will refer to the
2 individual pixels within the texel array as texels.

3 3. The texel array is "bound" to the polygons or surfaces. In other words, each
4 polygon is associated with a portion of the texel array that contains a description
5 of the appearance (e.g. color, brightness, saturation, a pattern of colors, etc.) that
6 that polygon will exhibit.

7 4. A 3D graphics pipeline then uses the texel array and computer model of the
8 surface to generate an image. This image is provided as another pixel array that is
9 displayed on a computer monitor or other output device.

10 While the above-mentioned process is adequate for some purposes, this process is
11 has some shortcomings when one tries to apply certain textures to the computer model of
12 the surface. For example, assume that instead of applying a simple texture contained in a
13 texel array to an object, one tries to apply a pixel array that is a photograph of an object
14 (e.g. a book on a table). The book is, of course, a three-dimensional object. If one
15 simply uses a 3D graphics pipeline to apply such a pixel array to the computer model of
16 the surface, the result will be a surface having a 2D image of a book thereon. In other
17 words, the 3D characteristics of the book will be lost. It would be desirable to preserve
18 the 3D characteristics of 3D objects depicted in a 2D pixel array when applying the pixel
19 array to a computer model of a geometric surface. This would result in the construction
20 of more realistic images to be displayed on a CRT.

21

22 **SUMMARY**

23 A method in accordance with our invention comprises:

- 1 a) providing a computer model of a surface;
- 2 b) providing a 2D image in the form of a pixel array;
- 3 c) providing additional intrinsic information about the 3D characteristics of the 2D
- 4 image;
- 5 d) binding the pixel array to the computer model of the surface; and
- 6 e) offsetting the surface within the computer model as a function of additional
- 7 intrinsic information.

8 In one embodiment, the surface of the computer model is offset in the Z direction.

9 In one embodiment, the pixel array contains a representation of a three
10 dimensional object. For example, suppose the pixel array contains a representation of a
11 book on a table. Books have a certain thickness. The computer model of the geometric
12 surface is modified in accordance with above-mentioned additional intrinsic information
13 pertaining to the 3D characteristics of the book. Accordingly, the portion of the
14 geometric surface where the book is to be rendered will appear displaced relative to the
15 rest of the geometric surface. This creates a “3D” appearance for the book.

16 In one embodiment, the pixel array and modified computer model of the
17 geometric surface are rendered into an image using a 3D graphics pipeline. The 3D
18 graphics pipeline typically receives as inputs information concerning the location and
19 orientation of the geometric surface in a “world coordinate system.” The 3D graphics
20 pipeline also receives as inputs information concerning where a “virtual observer” is to
21 observe the object being rendered, and where a “virtual lighting source” is to illuminate
22 the object. The 3D graphics pipeline takes this information and generates a second pixel
23 array corresponding to an image that is to appear on a computer display device such as a

1 CRT. As the orientation of the geometric surface is modified by the 3D graphics
2 pipeline, the 3D characteristics of the pixel array mapped onto the surface are preserved
3 because the computer model of the geometric surface has been modified in accordance
4 with those 3D characteristics. Thus, the resulting image generated by the pipeline will be
5 more realistic.

6

7 **BRIEF DESCRIPTION OF THE DRAWINGS**

8 Figs. 1A to 1E illustrate the operation of a 3D graphics pipeline.

9 Figs. 2A and 2B illustrate manipulation of a 2D image .

10 Fig. 3 is a simplified block diagram of a personal computer (PC) coupled to a
11 graphics controller with a 3D graphics pipeline.

12 Fig. 4 is a flow diagram of an improved method for generating 3D images from a
13 2D pixel array of data.

14 Fig. 5 is a flow diagram of an embodiment of the invention using a filter for
15 processing Z displacement values.

16 Fig. 6 is a flow diagram of a portion of an embodiment of the invention using an
17 image analyzer for generating Z displacement values.

18 Fig. 7 illustrates the manner in which an image filter is used to generate
19 displacement coefficient values.

20

21 **DETAILED DESCRIPTION**

22 As mentioned above, a method in accordance with our invention involves
23 modifying a computer model of a surface to offset portions of the surface in the Z

1 direction as a function of information contained in a pixel array, and then using the
2 modified computer model and pixel array to generate a more realistic 3D image.
3 Rendering the 3D image typically involves the use of a 3D graphics pipeline.
4 Accordingly, we will first explain how a 3D graphics pipeline is normally used.
5 Thereafter, we will describe its use during a method in accordance with our invention.

6

7 3D Graphics Pipelines

8 The 3D graphics pipeline referred to in this patent is a convention graphics
9 pipeline well known in industry, e.g. as described in Foley, et al., "Computer Graphics—
10 Principles and Practice", published by Addison Wesley in 1997. The graphics pipeline
11 can be implemented by a combination of hardware elements, known as accelerators, and
12 software, some of which is sometimes referred to as drivers. The partitioning between
13 hardware and software may vary, depending upon the CPU used and the graphics card in
14 the system, but the overall system performs the method described below. Portions of the
15 pipeline tasks can be performed by software, which is less expensive than hardware, but
16 in general slower than hardware solutions at the present time. The hardware and software
17 that perform the method described below are referred to simply as a pipeline, without
18 regard to the specific partitioning.

19 The following is a simplified, general description of 3D graphics pipelines. It is
20 not intended to describe any specific product (e.g. products mentioned later in this
21 patent). Rather, the following description is merely a general explanation of 3D graphics
22 pipelines to assist the reader's understanding.

23 Currently, graphics objects created using a 3D graphics pipeline can be described
24 as a set of geometric surfaces. One way of constructing a geometric surface in a graphics

1 pipeline is to create a “mesh” of “primitives.” A “primitive” is a small geometric surface
2 that can be defined by a set of vertices. For example, the primitive can be a polygon (e.g.
3 a triangle or quadrilateral) defined within the pipeline in terms of the locations (in x, y
4 and z coordinate space) of its corners or vertices. A set of several primitives is used to
5 define a larger 3D surface.

6 Instead of using primitives, such as polygons, some graphics pipelines can
7 process geometric surface areas defined in other ways, e.g. by mathematical equations.
8 This technique for defining geometric surface areas is called “implicit.” As explained
9 below, both techniques for defining such surface areas can be used to perform a method
10 in accordance with our invention.

11 For purposes of clarity of explanation, we will first describe a graphics pipeline
12 that processes geometric surface areas using triangular primitives. We will discuss other
13 types of graphics pipelines later on.

14 In this first example, a 3D graphics pipeline constructs a 3D image of an object
15 from a 2D texel array (typically called a “texture map”). Fig. 1A illustrates a 2D image 2
16 of a set of “textures.” (As will be explained below, this texture map is used to create the
17 image of an object—in this case, a house. Image 2 includes a portion 2a, which has the
18 appearance of bricks, portion 2b, which has the appearance of roof shingles, portion 2c,
19 which has the appearance of a door, and portion 2d which has the appearance of a
20 window.) 2D image 2 is stored in a digital memory in the form of an array of texels.
21 Each location in the memory stores a texel, which is one or more words of data indicating
22 the color, color saturation and brightness corresponding to that texel. The location of the
23 texels within the array is typically referred to as u, v coordinates (not to be confused with

1 the Y, U and V signal names used to describe certain video signals). The u, v coordinates
2 are similar to x, y coordinates of the Cartesian coordinate system. In Fig. 1A, the texel
3 array is an n by m array, where n and m are integers.

4 As mentioned above, Fig. 1A represents a texel array. Physically, the array
5 comprises data loaded into a memory.

6 The next step in the process is to provide or prepare a geometric surface. In this
7 example, the geometric surface is in the form of a mesh 4 of primitives 5 in three
8 dimensional space (Fig. 1B). In the case of Fig. 1B, the primitives are triangles, but other
9 types of polygons can be used. The mesh of primitives represents a three-dimensional
10 shape of an object O in 3D space (in the case of Fig. 1B, the shape of a house). The
11 position of each vertex of each triangle within mesh 4 is stored in a memory in the form
12 of x, y and z Cartesian coordinates, relative to the object. These coordinates are
13 sometimes referred to as model coordinates ("MC"). The process of preparing such a
14 mesh is well-known, and described in standard graphics libraries, such as Real 3D,
15 published by Real 3D, a Lockheed Martin Corporation, in 1996, and Direct 3D, published
16 by New Riders Publishing in 1997.

17 The mesh of Fig. 1B is not displayed as such. Rather, the mesh of Fig. 1B is a
18 representation of what is stored in a digital memory. Specifically, the memory stores the
19 locations, in terms of x, y and z coordinates, of each vertex within mesh 4.

20 The next step is to map or "bind" the two-dimensional texture map of Fig. 1A
21 onto mesh 4 of Fig. 1B. This is accomplished by mapping each triangle vertex to a
22 location in the texture map. In effect, a list of data points is prepared that associates each
23 vertex of mesh 4 to the u, v coordinates of a particular point (texel) in the texture map of

1 Fig. 1A. (The locations in the texture map to which the vertices are bound are sometimes
2 referred to as "control points.")

3 This portion of the process is roughly analogous to an upholsterer choosing a
4 piece of fabric, and binding it with a few nails to the corner of a couch being upholstered
5 (the nails are like control points). The upholsterer subsequently asks his apprentice to
6 finish attaching the fabric to the couch. In this case, the 3D graphics pipeline finishes the
7 task instead of an apprentice.

8 Figs. 1A and 1B describe the process by which one texture map (Fig. 1A) is
9 mapped onto one mesh 4 representing one object O. A graphics pipeline can, and often
10 does, map one or several texture maps onto the same or several different objects.

11 The next step in the process is to set up a "world coordinate model" of the various
12 objects to be displayed. This requires establishing a position and directional orientation
13 for each object to be displayed. For example, supposing that instead of a house there are
14 to be two objects to be viewed: a tetrahedron T and a cube C (Fig. 1C). During this
15 portion of the process the pipeline is instructed that cube C is to be facing in a certain
16 direction, and is to be located partially in front of tetrahedron T relative to a certain frame
17 of reference. Again, the structure of Fig. 1C is not displayed per se. Rather, the graphics
18 pipeline sets up processing of the model coordinates in accordance with the parameters of
19 the position and orientation of the object.

20 The next step is to select a frame of reference. For example, it might be decided
21 that the "viewer" will want to observe the objects from a position corresponding to a
22 corner of the world coordinate model (e.g. position P in Fig. 1D). Thus, a virtual
23 viewpoint, viewing direction and aperture will be selected. The parameters associated

1 with this "viewer" define the screen coordinate (SC) system. Further, it might be decided
2 the viewer will observe these objects with a light source located at a position L. The
3 graphics pipeline will set up another processing pipe to process the world coordinate data
4 into the screen coordinate data which will cause a computer screen to display the image
5 as it would be perceived by the observer at position P (e.g. the image of Fig. 1D). In
6 other words, the computer screen will provide an image of tetrahedron T and cube C as
7 they would be observed by a viewer if he were standing at position P, and a light source
8 were present at location L. This image will be provided initially as a pixel array in a
9 frame buffer and then displayed by the computer screen. The image in the frame buffer
10 is refreshed, i.e. regenerated according to the specifications programmed into the
11 pipeline, typically at about 50 to 120 times per second. There are many different
12 methods for optimizing the pipeline, and minimizing the time spent processing the
13 invisible parts of the objects, such as the backside of cube C facing away from the
14 viewer. Such details are well-known to those skilled in the art, and will not be discussed
15 in detail here.

16 During the above-described process constructing the pixel array and providing it
17 in the frame buffer, the pipeline a) fetches the portion of texture map 2 "tacked" to the
18 vertices of mesh 4 (and therefore stretched over each triangle); b) determines how and
19 where that portion of the texture map should appear, given the orientation of the triangles
20 relative to the viewer and the location of the light source; and c) constructs the
21 appropriate bit map pixel array for storage in the frame buffer. The contents of this frame
22 buffer are then displayed as an image on a computer screen.

1 Thereafter, the 3D graphics accelerator permits one to manipulate the displayed
2 objects in any desired manner. For example, if one wants to rotate the image of
3 tetrahedron T by 45° (Fig. 1E), the 3D graphics accelerator facilitates this manipulation.
4 This is accomplished by providing a new set of parameters in the world coordinate model
5 for the graphics pipeline indicating the new position and orientation for tetrahedron T.
6 After this occurs, the next time the graphics pipeline regenerates the image stored in the
7 frame buffer, the regenerated image will reflect this rotation of tetrahedron T.

8 Similarly, suppose that it is desired to display what would appear to the viewer if
9 he took ten steps forward from his location at position P. The next time the graphics
10 pipeline regenerates the image, it will generate and store another pixel array in the frame
11 buffer corresponding to what would appear to such a viewer, and this pixel array is
12 provided as another image on the computer screen.

13 It is thus seen that the graphics pipeline is extremely useful in applications such as
14 video games, where it is desired to simulate what would appear to a game player if he
15 were wandering past a set of objects.

16 As mentioned above, some graphics pipelines create models of geometric surfaces
17 using an implicit technique. These surfaces are often described as a function of the
18 position coordinates, i.e. $f(x,y,z)$, or can also contain some vertices. Control points and
19 additional formulas associated with such surfaces are used to bind a digital texel array
20 (e.g. an array as shown in Fig. 1A) to the implicitly defined surface, and the process
21 proceeds as described above. The major difference is that instead of defining surface
22 areas in terms of primitives with vertices, the surface areas are defined in terms of
23 mathematical equations.

1

2 Manipulation of 2D Images

3 Our previous U.S. Patent application serial no. 09/344,442 (incorporated herein
4 by reference) discloses a method of applying a 2D image onto a geometric surface. A
5 method in accordance with this application begins with the step of obtaining a two-
6 dimensional digital image (e.g. image 10 in Fig. 2A). This step can be performed, e.g.,
7 by scanning an image such as a photograph or other picture using a conventional digital
8 scanner. The digital image can also be obtained from a conventional digital camera. The
9 image can also consist of digital video image, e.g. out of a live or stored video stream,
10 which is basically a fast succession of 2D images. However, any other source of a 2D
11 digital image can be used. As mentioned above, the digital image is typically stored in a
12 memory as an array of digital values. In one embodiment, the digital values are in a
13 compressed form, e.g. using a compression technique such as MPEG1 or MPEG2 or
14 other formats. In the case of compressed digital values, they must first be decompressed
15 prior to processing. Also, scanned images or digitized images from any source such as
16 cable TV, an antennas, cameras, etc. can be used.

17 As mentioned above, for the case of video images, dozens of frames per second
18 comprising millions of pixels per second must be processed. Standard graphics pipelines
19 can be used to process frames of data sufficiently fast to process video images.

20 Any type of memory can be used to store the digital 2D image, e.g.
21 semiconductor memories (SRAMs, DRAMs or other semiconductor memories), a
22 magnetic memory (e.g. a hard disk, a floppy disk, magnetic tape, or magneto-optic disk),
23 or other type of memory device (e.g. an optical disk). The pixels corresponding to the

1 stored image can be stored in terms of RGB values (e.g. the strength of the red, green and
2 blue components of the pixel color), YUV values or other values. (For YUV values, Y
3 corresponds to the amplitude or brightness of the pixel value, U corresponds to the color
4 and V corresponds to the saturation.) The pixel values can be encoded in other ways as
5 well. Depending on the situation, a conversion may be required before further
6 processing.

7 Next, a 3D graphics pipeline is set up. This is accomplished by providing
8 instructions to the 3D graphics pipeline as to what is to be done with the data that is to be
9 provided. Setting up graphics pipelines per se is well known in the art, e.g. as described
10 in the Microsoft Direct 3D SDK (software developer kit) or Direct 3D.

11 Thereafter, a computer model of a planar geometric surface is generated. This
12 computer model can comprise a set of primitives, e.g. polygons such as triangles. In
13 another embodiment, the computer model can comprise an implicit description of a flat
14 geometric surface. This implicit description is typically a mathematical function (e.g. a
15 function of x, y and z) as described above.

16 For the case in which the planar geometric surface comprises a mesh of
17 primitives, the number and shape of primitives and the type of primitives can vary. Fig.
18 2B illustrates a mesh 12 that can be used to practice a method in accordance with our
19 invention. Mesh 12 is similar to mesh 4 described above. However, unlike mesh 4, all of
20 the vertices of mesh 12 are coplanar (or substantially coplanar). In one embodiment,
21 mesh 12 comprises about 5000 triangles, which would be acceptable for processing a
22 video image. Of course, other numbers of primitives could be used.

1 After constructing the planar geometric surface (e.g. mesh 12), image 10 is
2 mapped, or bound, onto the flat geometric surface. This is accomplished in the following
3 way. For the case in which the flat geometric surface is a mesh such as mesh 12, each
4 vertex of the flat geometric surface (e.g. the triangle vertices) is associated with an image
5 pixel location (i.e. control point). Thus, each control point is associated with a texture
6 coordinates (u, v) corresponding to a pixel. A table of data listing each vertex and its
7 associated u, v texture space coordinates is set up. This is called "binding." (See Kamen,
8 IEEE Computer Society, IEEE Computer Graphics and Applications, Jan.-Feb. 1997,
9 Vol. 17, No. 1.) For the case in which an implicit technique is used to define the flat
10 geometric surface, control points within the implicitly defined surface are bound to pixel
11 array coordinate space (u, v coordinates) in a manner analogous to the triangles discussed
12 above.

13 After image 10 is mapped into mesh 12, the object can be manipulated by
14 manipulating the world coordinates. The world coordinates describe where in the x, y, z
15 space the textured plane is to appear, and what its orientation will be (i.e. what angle it
16 should be held at with respect to the viewer). In addition, the screen coordinates for the
17 object can be changed. As a result, when the 2D textured image is finally prepared, it can
18 be prepared in a manner that reflects the desired manipulation. For example, it can be
19 rotated about any axis, magnified, shrunk, etc.

20 After establishment of the world coordinate model and screen coordinate model,
21 the pipeline prepares an array of pixels in the output frame buffer (OFB), including pixels
22 showing the manipulated textured mesh 12. The array of pixels in the OFB is displayed
23 on a CRT or other type of screen.

1 One can manipulate the video image by, for example, changing the world
2 coordinate parameters, e.g. telling the pipeline to tilt the video image about any axis
3 (including an axis perpendicular to the screen or in the plane of the screen). Thus, when
4 the pipeline regenerates the pixel array in the OFB, the regenerated video image will
5 appear tilted about the selected axis. Since the pipeline will regenerate the image at a
6 preprogrammed rate according to the system used, live video will appear as live video.
7 That is because every time a new pixel array is generated, the texture map, which
8 contains the incoming video frame buffer, is reread and put through the pipeline. Since
9 the texture mapping process also contains features for pixel interpolation, an automatic
10 resolution adaptation occurs.

11 One can bend or warp the image by moving the vertices about which the image is
12 mapped. Thus, one can alter the flat geometric plane of Fig. 2B to thereby warp the
13 image. When the pipeline regenerates the pixel array in the frame buffer, the image will
14 appear warped.

15 One can move the vertices so that mesh 10 becomes a cylinder. When the
16 pipeline regenerates the pixel array in the frame buffer, the image will appear wrapped
17 around a cylinder. (Of course, the mesh 10 can be altered into other shapes, and the
18 image would be wrapped around the other shape.) These modifications can be done at a
19 speed that creates the impression in the viewer that the image is being wrapped or warped
20 gradually.

21 One can magnify or shrink images by moving vertices away from or closer to
22 each other, or moving the image closer or further from the viewer in the world coordinate
23 system, or by re-parameterizing the model coordinate to world coordinate conversion.

1 One can cause the image to "melt" by causing the verticies to drop at different
2 rates.

3 The number of 3D manipulations possible is only limited by the energy of a user
4 to try them all out.

5

6 Hardware and Software for Practicing An Embodiment of a Method In Accordance With
7 Our Invention

8

9 One embodiment of our invention can be practiced using a PC having the
10 following:

11 1. A CPU such as a Celeron or Pentium, e.g. as manufactured by Intel, or a K6
12 processor, e.g. as manufactured by Advanced Micro Devices.

13 2. 32 MB of memory or greater.

14 3. A 3D HW adapter. This is a type of graphics card currently available on the
15 market. The 3D HW adapter should have 4 MB of memory (preferably 8 MB)
16 and an advanced graphics port (AGP) interface. (An AGP interface is a type of
17 bus standard that is well-known in the art.) Alternatively, a peripheral connection
18 interface ("PCI") can be used in lieu of a AGP. The PCI is a type of bus standard
19 that is well known in the art. Examples of appropriate 3D HW adapters include
20 the TNT-2 available from Riva, the ATI Rage 128, the Matrox G400, the Trident
21 Blade 3D and the S3 Savage.

22 4. The operating system can be Windows 95, Windows 98, Win2000, or any other
23 operating system that supports direct 3D. The Windows operating system
24 includes a standardized platform called Direct X for Windows.

1 In one embodiment, a user sets up the flat geometric surface (for example, a
2 triangle mesh) in the Direct 3D windows environment. The set of instructions is then
3 provided to the graphics pipeline, which finishes the rendering process. However, in
4 another embodiment, the PC comprises a bypass mechanism that permits one to access
5 the hardware accelerator directly using a software interface provided by the graphics card
6 manufacturer.

7 Fig. 3 is a block diagram of a computer system 50 for performing a method in
8 accordance with our invention. Referring to Fig. 3, system 50 comprises a CPU 52, e.g. a
9 Pentium II class CPU, comprising a cache memory 52a, a core 52b and an internal bus
10 52c for facilitating communication between core 52b and cache 52a. Core 52b
11 communicates via a CPU bus 54 to a system controller 56. System controller 56
12 communicates with the system memory 58 via a memory bus 60. System memory 58
13 includes a first portion 58a which stores system memory programs and a second portion
14 58b that stores the texture maps such as described above.

15 Also included in system 50 is a PCI bus 62 for facilitating communication
16 between system controller 56 and I/O devices 64, 66 and disk drive 68. I/O device 64
17 can be any type of I/O device. In one embodiment, I/O device 66 is a video capture card
18 with a driver. Data from the video capture card is either loaded by DMA (direct memory
19 access) or CPU 52 into a frame buffer, typically within main memory 58. However, the
20 frame buffer may be in other memories within system 50.

21 System 50 also includes an AGP graphics controller 70 comprising a 3D
22 accelerator. In one embodiment, AGP graphics controller 70 communicates with system
23 controller 56 via an AGP bus 72. In an alternative embodiment, AGP graphics controller

1 70 can communicate with system controller 56 via PCI bus 62 (e.g. as shown in phantom
2 in Fig. 3).

3 Graphics controller 70 uses its own local memory 74 to generate and store pixel
4 arrays to be displayed on a video display unit 76.

5 It is emphasized that system 50 is only one example of a system that performs a
6 method in accordance with our invention. Other hardware can be used as well.

7

8 First Embodiment of 3D Model Creation Based on a 2D Image

9 Referring to Fig. 4, a method in accordance with the present invention for creating
10 a 3D model based on a 2D image begins with the step of providing a 2D image in the
11 form of an array of pixels 100 and a set of geometric surfaces 102. The geometric
12 surfaces can be a grid of polygons or they can be implicitly defined. As will be explained
13 in greater detail below, it would be desirable, but not necessary, to have an established
14 relationship between the grid of polygons and the 2D image. (In other words, it would be
15 desirable to have a pattern in the polygons that matches the 2D image to facilitate binding
16 the pixels to the polygons.) Array of pixels 100 can be provided from any appropriate
17 source, e.g. as described above and in our '442 application.

18 Together with the 2D image (or array of pixels 100) additional information about
19 image-related "intrinsic" 3D geometry is provided. For instance, this information can be
20 the luminance of the image, its hue, saturation, or an external "depth map" which
21 describe pixels' distance from eye point in a normalized form. As described in greater
22 detail below, this additional information is used to modify geometric surfaces 102 so that
23 the geometric surfaces reflect the 3D aspects of the image depicted in pixel array 100.

1 This additional information is hereafter referred to as a “Z array” 104. Z array 104 may
2 have a size or resolution that is different from pixel array 100. Z array 104 comprises
3 data in an x, y coordinate system. The data within Z array 104 is hereafter referred to as
4 a “displacement coefficient.” Z array 104 can be stored in the same type of memory
5 devices as pixel array 100.

6 Z array 104 is bound to geometric surfaces 102 (step 106 of Fig. 4). This is done
7 in a manner similar to the binding of a prior art texel array to a geometric surface. Thus,
8 each vertex within geometric surface 102 is associated with a location (in terms of x, y
9 coordinates) within Z array 104.

10 If a Z array data location bound to a vertex of geometric surface 102 is a control
11 point, then the displacement coefficient corresponding to that Z array location is assigned
12 to that vertex. That vertex’s Z coordinate is then displaced by a distance that is
13 proportional to the displacement coefficient. In other words, the vertex’s coordinates are
14 modified, and the vertex is moved in the Z direction by a distance proportional to the
15 displacement coefficient. This typically occurs within the graphics pipeline, although it
16 can also be done by CPU 52.

17 If the Z array location bound to a vertex of geometric surface 102 is not a control
18 point then the Z position for that vertex is interpolated as a weighted sum of the values of
19 its neighbor control points. In other words, the position (in the Z direction) of each non-
20 control point vertex is recalculated by interpolation as a function of the surrounding
21 control point Z positions. Edges can be handled by mirroring, banding, or wrap-around
22 techniques. If there are no control points, the vertex is simply offset in accordance with
23 the Z array value for the location bound to that vertex.

1 Thereafter, pixel array 100 is bound to modified geometric surface 102 in a
2 manner similar to the manner in which conventional texel arrays are bound to a geometric
3 surface (step 110 in Fig. 4).

4 Thereafter, the parameters for the 3D graphics pipeline are established. For
5 example, information 112 is provided corresponding to how the modified geometric
6 surface is to be positioned in the world coordinate system, the location of a virtual
7 viewer, and a location for a virtual lighting source. A 3D graphics pipeline takes this
8 information and uses it to render an image (step 114 of Fig. 4) and provide that image in
9 the form of a pixel array 116 in a frame buffer. The contents of the frame buffer are then
10 displayed on a device such as a CRT.

11

12 **Second Embodiment Including Image Filter 202**

13 In one embodiment, Z array 104 is smoothed or otherwise modified by a filter 202
14 before geometric surface 102 is modified (Fig. 5). In this embodiment, image filter 202
15 receives as inputs Z array 104 and a set of filter parameters 204, and provides in response
16 thereto filter output array 206. Instead of using Z array 104 to modify the Z coordinates
17 of geometric surface 102, in this embodiment filter output array 206 is used to modify the
18 Z coordinates of geometric surface 102. The filter parameters include a) aperture size, b)
19 aperture center, c) aperture form, and d) aggregation scheme. The image filter essentially
20 applies a mathematical function that takes as its inputs the data within Z array 104 and
21 provides an output value for each data point within the array.

22 Merely by way of an example, one type of filter might have an aperture that
23 includes each Z data point surrounding the aperture center. Filter 202 analyzes each data

1 point within Z array 104 in sequence. For each Z data point being analyzed, the input
2 values on which the image filter operates include all of the Z data points within the filter
3 aperture. For example, suppose that a) the filter is analyzing the Z data point at
4 coordinate 10, 10; and b) the filter's aperture includes all of the Z data points surrounding
5 the Z data point at coordinate 10, 10. Thus, the image filter takes as inputs the eight Z
6 data points surrounding Z data point 10, 10 (i.e. the data points at coordinates 9, 9; 9, 10;
7 9, 11; 10, 9; 10, 11; 11, 9; 11, 10; and 11, 11). The image filter thus has an aperture that
8 causes the filter to evaluate eight data points. The term "aperture form" refers to the
9 shape of the aperture. In this example, the aperture form is a square containing the eight
10 Z data points surrounding the aperture center.

11 The aggregation scheme of the filter is the mathematical function or formula by
12 which the filter takes all of the values corresponding to the data points within the aperture
13 and generates a single number as the filter output. One example of a filter simply
14 calculates an average of the values of the eight above-mentioned data points. The filter
15 then calculates a normalized value corresponding to this average. The normalized value
16 is between zero and one, where one is the maximum possible average intensity, and zero
17 is the minimum possible average intensity.

18 For the case of data points located at the edges of Z array 104, the filter can
19 calculate its output array 206 by mirroring, banding or wrap-around techniques.
20 (Mirroring merely means that the displacement coefficient for data points at an edge of
21 the array are calculated by assuming that any locations within the filter aperture but
22 outside the Z array boundary are the mirror image of the data points inside that boundary.
23 Wrap-around means that the filter assumes that any locations within the filter aperture to

1 the left of the Z array boundary equal the values at the far right edge of the Z array.
2 Similarly, any locations within the filter aperture above the top array boundary are treated
3 as equaling the values at the very bottom of the array. Banding assumes that any values
4 within the aperture past an edge of the array have values identical to the column of data
5 points immediately adjacent that edge. In one embodiment, a Bartlett filter is used for
6 performing this function.)

7

8 Generation of Z array 104

9 Image array 100 and Z array 104 can be generated using any of a number of
10 techniques. In one embodiment, image array 100 and Z array 104 can be provided using
11 cameras. The camera used to generate Z array 104 processes information such as
12 luminance, and the Z array data is based on the output of that camera.

13 In another embodiment, Z array 104 can be provided by an image analyzer 150
14 (Fig. 6). Image analyzer 150 performs two functions. First, analyzer 150 selects the Z
15 array 104 values from a set of possible sources or generates the Z array 104 values using
16 an algorithm, e.g. as described below. Second, analyzer 150 selects the control points
17 within 2D pixel array 100. Image analyzer 150 is either controlled interactively by a
18 human operator or non-interactively in accordance with a pre-defined algorithm. Three
19 examples of image analyzers are described below:

20

21 Image Analyzer Example 1

22 In this example, 3D information in Z array 104 is a function of luminance
23 information contained within image pixel array 100. Geometric surface 102 is a flat

1 geometric surface comprising a uniform grid of 50 by 50 triangles. The position of each
2 vertex within this grid is matched with a corresponding location of a pixel within array
3 100 based on the geographic location of the vertex within the grid and the location of the
4 pixel within array 100. The pixels within array 100 that are matched with each vertex
5 constitute control points.

6

7 Image Analyzer Example 2

8 3D information is completely described in a gray scale image (depth map) which
9 has the same size as an original image. (The gray scale image is made using a camera
10 equipped with depth perception.) The control points are chosen using the technique
11 described for Example 1 above.

12

13 Image Analyzer Example 3

14 The data within Z array 104 is computed as an arithmetic mean of all red and
15 green color values for each pixel within array 100. Control points are chosen using the
16 method described in Example 1.

17 Other image analyzers are well-known in the art. Examples of appropriate image
18 analyzers include outline filters, contrast filters and functional derivatives of the original
19 image or a second, user supplied image. For the case in which image analyzer 150
20 generates Z array 104 as a function of pixel array 100, the image analyzer function (as
21 well as the filter 202 function) can be performed by CPU 52. CPU instructions pertaining
22 to the image analyzer function can be present in disk 68 or other mass storage device.

1 Alternatively, the image analyzer function can be implemented with a dedicated hardware
2 accelerator or a combination of hardware and software.

3 Fig. 7 shows an example of a method in accordance with the present invention. A
4 2D image 300 and a Z array 304 corresponding to an object 305 (a square) is provided. Z
5 array 304 includes control points 304a to 304d. In corner area 303, a vector forest (i.e. a
6 forest of vectors, each vector corresponding to a displacement coefficient for a given
7 location within Z array 304) is shown, that would occur for "natural data images", such
8 as photographs. (In other words, for certain types of images, such as images that might
9 be contained in a photograph, the displacement coefficient might vary from one pixel to
10 its neighbors.) However, most vectors are very similar, e.g. grass in the background of a
11 picture has many features, but is over all seen as "only" a texture. Hence an average
12 texture vector 313 corresponding to the average of the displacement coefficients in a
13 region of the pixel array can represent the vector forest. (The vectors associated with
14 pixels in the rest of the image are not shown, to avoid obscuring the drawing.) Lines 309
15 are shown in Fig. 7 to help visualize the manner in which the polygons are displaced, and
16 do not exist in the image being rendered.

17 Returning now to object 305, its points 302a to 302d are bound to control points
18 which are displaced in accordance with vectors 312a to 312d, making object 305 a raised
19 object in 3D, hence adding the third dimension based on the data in Z array 104. (As
20 mentioned above, the data within Z array 104 can be a function of color, texture, or other
21 parameters of pixel array 100.) This is accomplished by modifying geometric mesh 314
22 as shown in Fig. 7. When rendered as an image, square 305 will be raised in the z
23 direction.

1 It will be appreciated that modifying the position of the control points in the Z
2 direction will cause the image being rendered to be more realistic. It is often the case that
3 a change in color in a photographic image corresponds to an object having a certain
4 thickness. By modifying the control points in the Z direction in accordance with that
5 change of color, the resulting mesh of polygons will more accurately match the object
6 whose image is being rendered. If the 3D graphics pipeline is called upon to prepare an
7 image of that object from a different angle, that Z direction displacement will be
8 preserved, and more accurate images of the object can be prepared.

9 The image filter may not always displace the colored object in the Z direction by
10 an appropriate amount. For example, an object with a colored concave section may be
11 inadvertently portrayed as a convex object, or vice versa. Accordingly, in some
12 embodiments, a human operator may monitor or modify the manner in which the image
13 is being modified in order to ensure that the image is being adjusted properly.

14 After the geometric adjustment of the object, an image is rendered using a 3D
15 graphics pipeline, which prepares a pixel array in a frame buffer memory corresponding
16 to the image. The image stored in the frame buffer memory is then displayed by a device
17 such as CRT.

18

19 Applications for the Invention

20 There are many applications for our invention. One application is to render 2D
21 images into 3D objects, without requiring prior construction of a correct geometric
22 surface, which can be an arduous task. Rather, one skilled in this art, using the present
23 invention, can simply select a few control points, and use the Z array to handle the rest.

1 In case of a house, or a simplified cube, he or she would choose exterior boundary points,
2 such as the corners on top of the house, or on the roof, and based on the color differences
3 and other aspects, such as brightness, etc., the Z array would used to raise the house from
4 the surrounding land, much like a laborious process of making a geometric surface would
5 do. Then the 2D image is bound and “stretched” onto that geometric surface, thereby
6 generating a 3D representation of a house, from a 2D image, using very few (if any)
7 manual steps, and using a 3D pipeline in a novel manner.

8 While the invention has been described with respect to a specific embodiment,
9 those skilled in the art will realize that changes can be made in form and detail without
10 departing from the spirit or scope of the invention. For example, instead of moving the
11 geometric surface in the Z direction, one can move the geometric surface in another
12 direction. Accordingly, all such changes come within the invention.

1

2 We claim:

3 1. A method for forming an image comprising the acts of:
4 providing an array of pixels;
5 providing a mesh of geometric surfaces;
6 providing displacement information;
7 modifying the position of said geometric surfaces as a function of said
8 displacement information;
9 binding said mesh of geometric surfaces to said array of pixels; and
10 rendering an image based on said array of pixels and said modified geometric
11 surfaces.

12

13 2. Method of claim 1 wherein said displacement information is in the form of an
14 array of displacement values, said method further comprising the step of subjecting said
15 displacement values to a filtering step.

16

17 3. Method of claim 2 further comprising the step of applying an image analyzer to
18 said pixel array to generate said displacement values.

19

20 4. Method of claim 3 wherein said image analyzer is used to bind said pixel array to
21 said geometric surfaces.

22

1 5. Method of claim 3 wherein vertices ²⁷ of said geometric surfaces corresponding to
2 control points within said pixel array are offset by a value proportional to said
3 displacement values.

4

5 6. Structure comprising:

6 a first memory for storing the location of one or more geometric surfaces;
7 a second memory for storing a pixel array, the pixels in said pixel array depicting
8 an image;

9 a third memory for storing information concerning 3D characteristics of said
10 image;

11 a circuit for binding said pixel array to said one or more geometric surfaces; and
12 a circuit for altering the location of said one or more geometric surfaces as a
13 function of the information in said third memory.

14

15 7. Structure of claim 6 wherein said first, second and third memories comprise
16 locations within the same memory device.

17

18 8. A method for forming an image comprising the acts of:
19 providing an array of pixels;
20 providing a computer model of a geometric surface;
21 providing displacement information;
22 modifying the position of said geometric surface as a function of said
23 displacement information;

1 binding said geometric surface to ²⁸said array of pixels; and
2 rendering an image based on said array of pixels and said modified geometric
3 surface.

4

- 5 9. A method for forming an image comprising the acts of:
 - 6 providing a first image;
 - 7 providing a set of one or more geometric surfaces;
 - 8 providing displacement information;
 - 9 modifying the position of said one or more geometric surfaces as a function of
 - 10 said displacement information;

11 binding said one or more geometric surfaces to said first image; and
12 rendering a second image based on said first image and said modified geometric
13 surfaces.

14

15 10. A method for forming an image comprising the acts of:

16 providing a first image;

17 providing a computer model of a geometric surface;

18 providing displacement information;

19 modifying the position of said geometric surface as a function of said

20 displacement information;

21 binding said geometric surface to said first image; and

22 rendering a second image based on said first image and said modified geometric
23 surface.

1
2 11. Method of claim 1, 8, 9 or 10 wherein said binding is performed said modifying.

3

4 12. Method of claim 1, 8, 9 or 10 wherein said modifying is performed before said
5 binding.

6

7 13. Structure comprising:

8 a first memory for storing the location of one or more geometric surfaces;

9 a second memory for storing a first image;

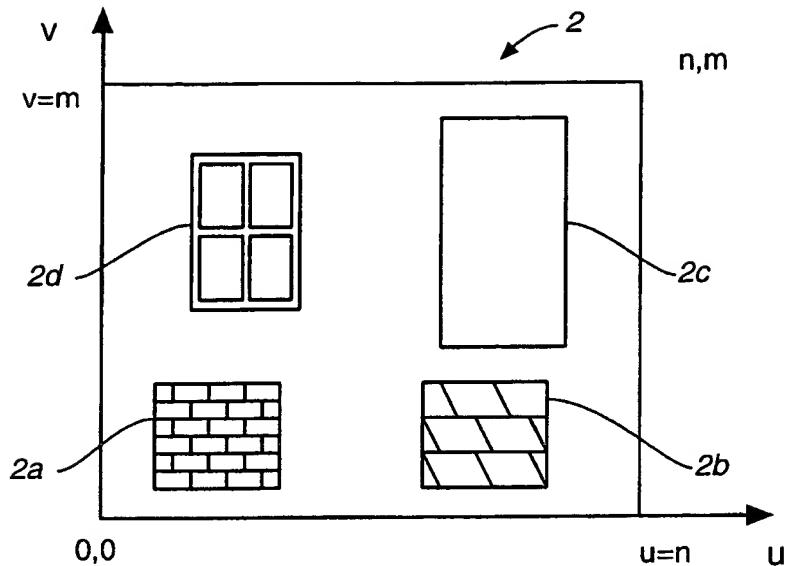
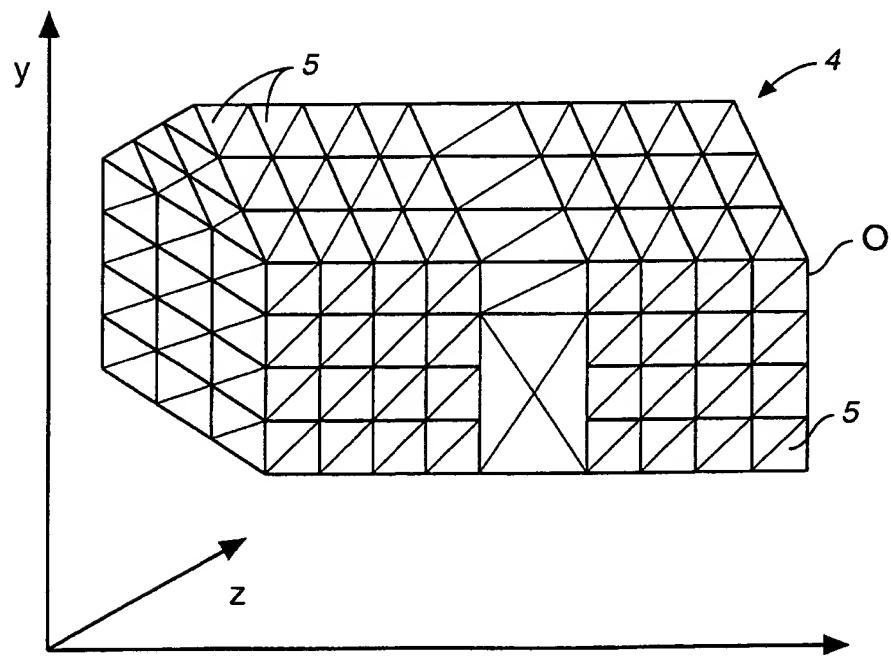
10 a third memory for storing information concerning 3D characteristics of said first
11 image;

12 means for binding said first image to said one or more geometric surfaces; and

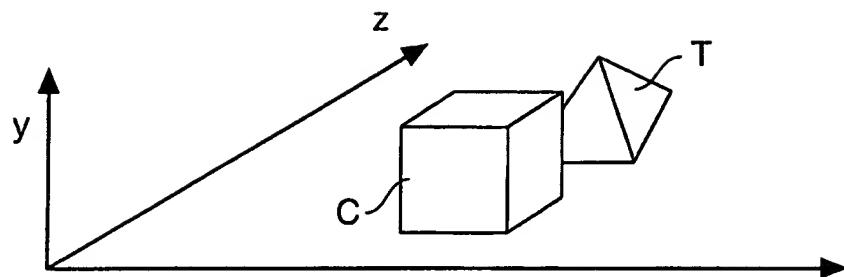
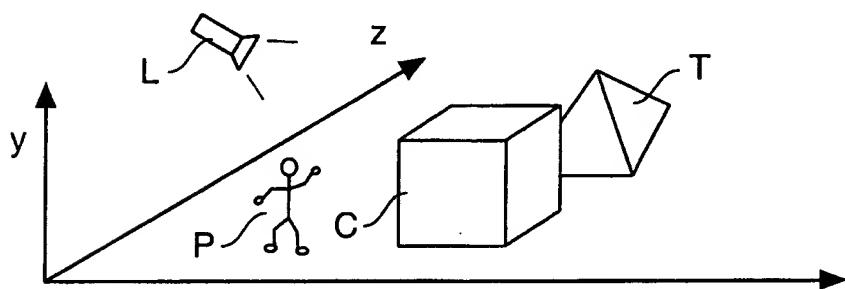
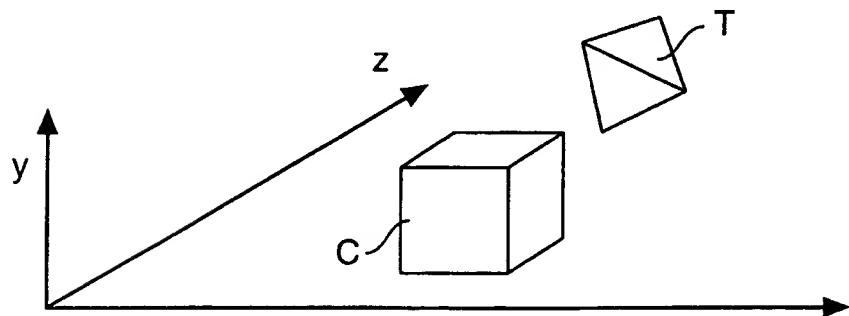
13 means for altering the location of said one or more geometric surfaces as a

14 function of the information in said third memory.

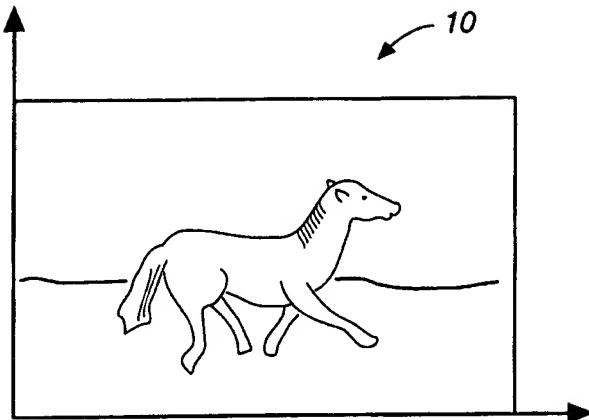
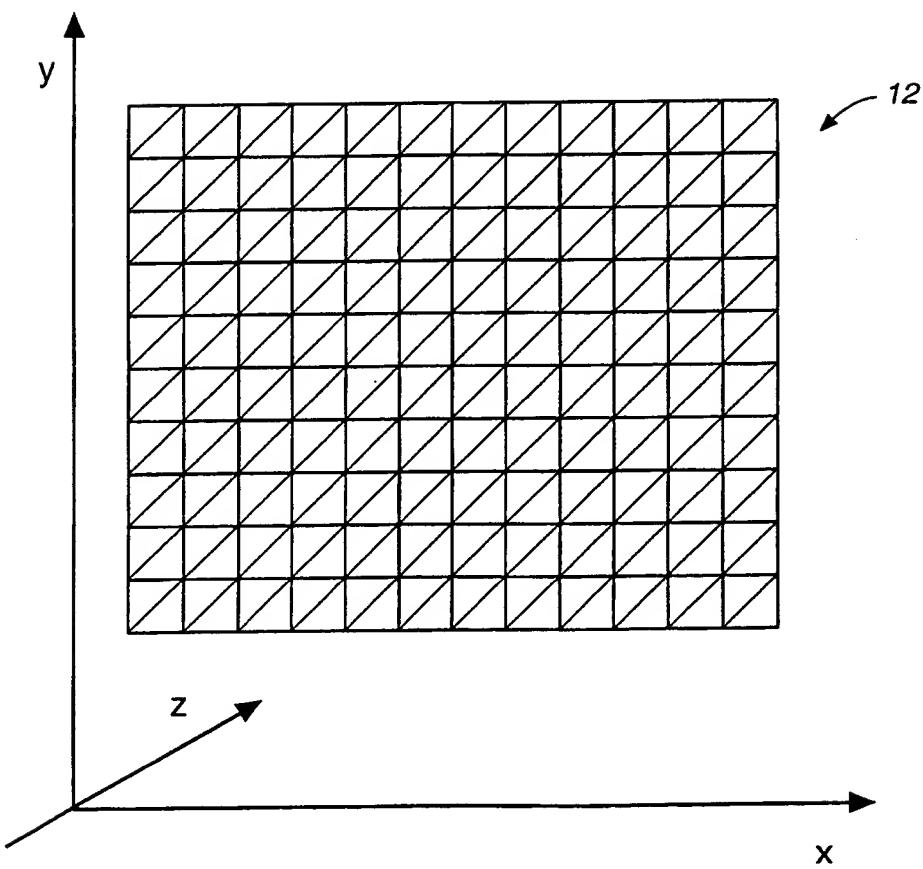
1 / 6

**FIG. 1A****FIG. 1B**

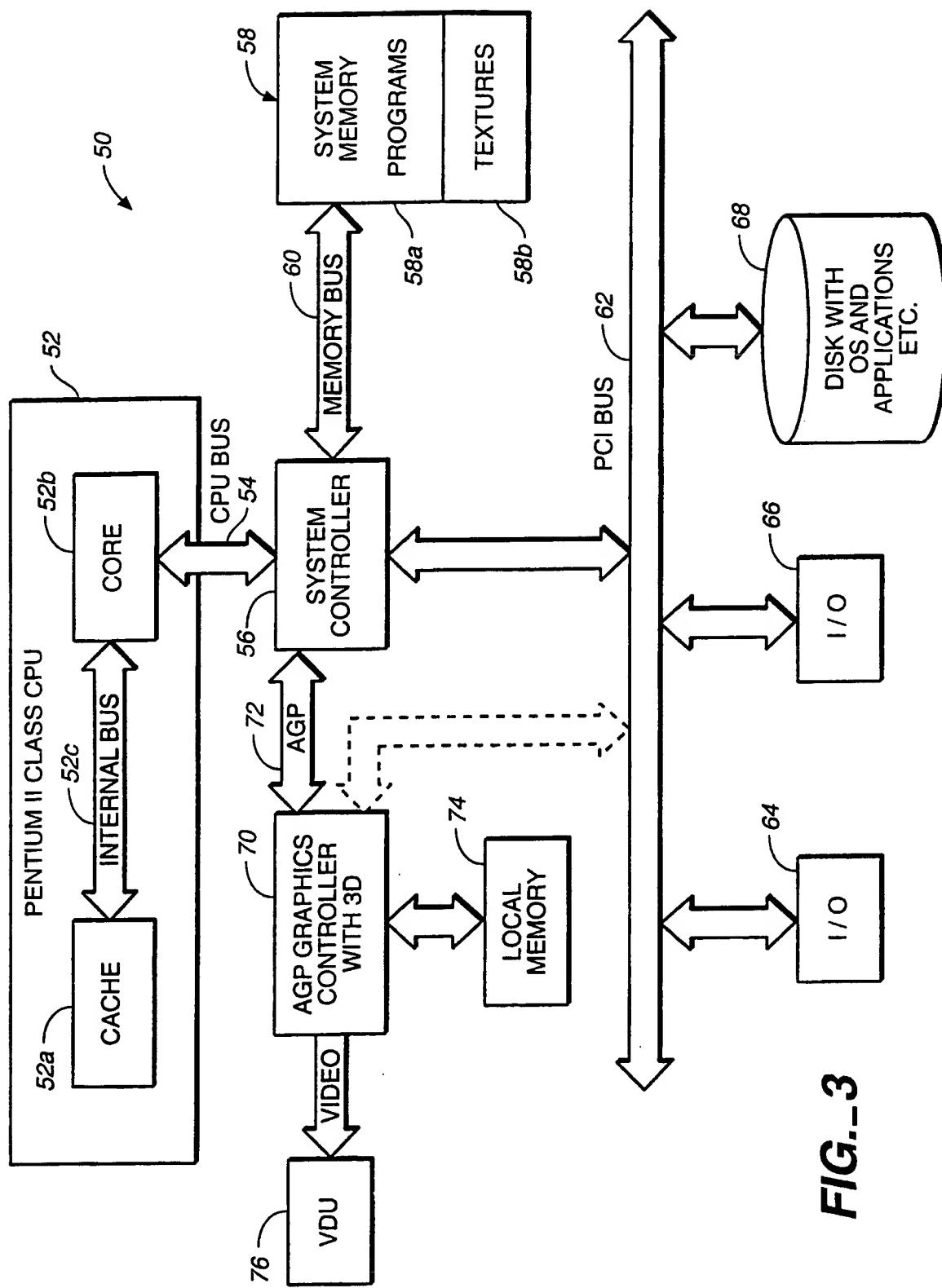
2 / 6

**FIG._ 1C****FIG._ 1D****FIG._ 1E**

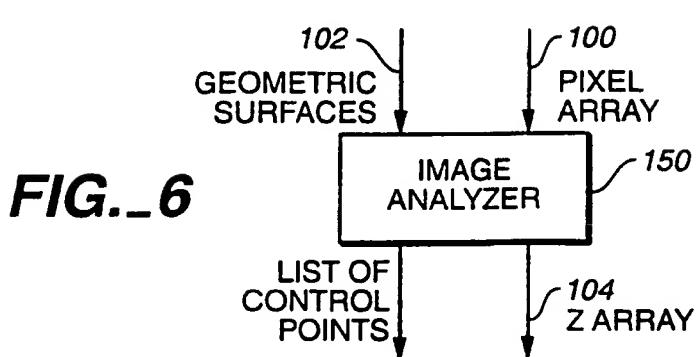
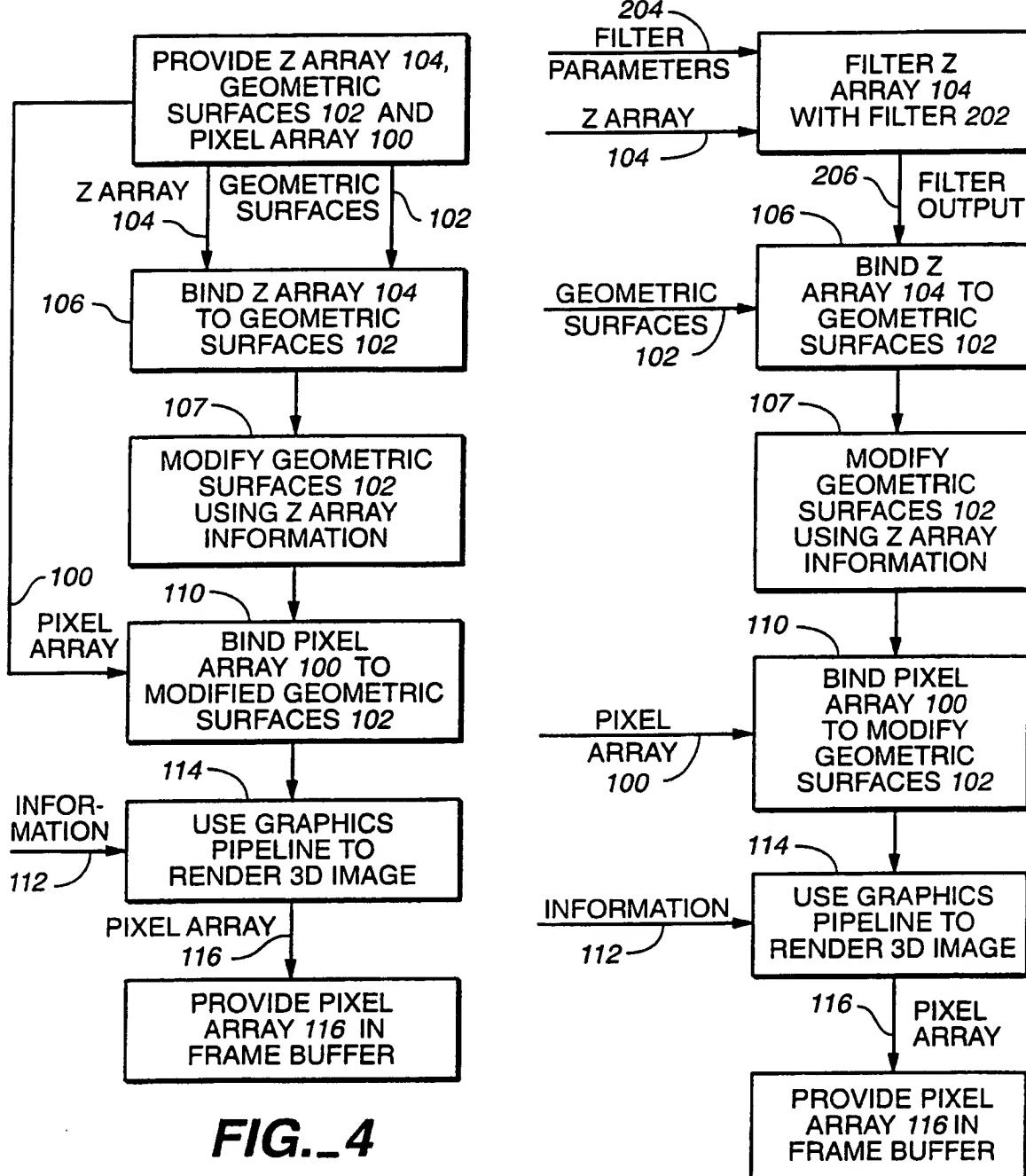
3 / 6

**FIG._2A****FIG._2B**

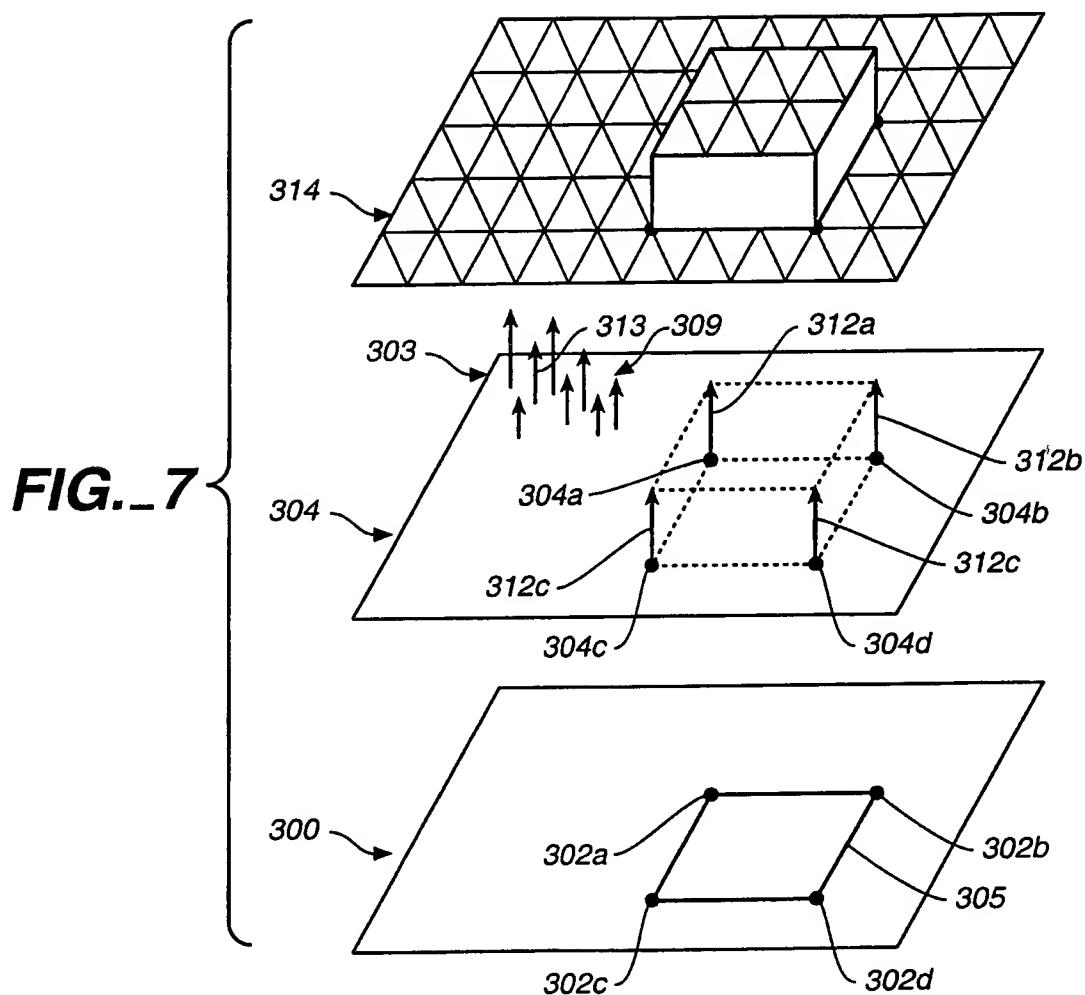
4 / 6

**FIG.-3**

5 / 6

**FIG. 6**

6 / 6



INTERNATIONAL SEARCH REPORT

International application No.

PCT/US00/02786

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06T 11/40

US CL : 345/425, 430

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/425, 430

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,838,331 A (DEBRY) 17 November 1998; col. 3, lines 31-36; col. 10, lines 16-19; col. 5, line 62 through col. 6, line 3; col. 5, lines 26-32; col. 3, lines 30-35.	1-13
Y	US 5,745,667 A (KAWASE et al.) 28 April 1998, col. 8, line 60 through col. 9, line 4; col. 5, line 62 through col. 6, line 3; col. 11, line 9-36; col. 11, lines 24-55; col. 10, lines 4-39.	1-5, 8-13
Y	US 5,566,285 A (OKADA) 15 October 1996; col. 2, lines 41-43; col. 2, lines 47-50; col. 7, lines 30-36; col. 2, lines 41-60.	6, 7

 Further documents are listed in the continuation of Box C. See patent family annex.

Special categories of cited documents:	*T*	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"		document defining the general state of the art which is not considered to be of particular relevance
"E"	*X*	earlier document published on or after the international filing date
"L"		document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O"	*Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"P"	*&*	document published prior to the international filing date but later than the priority date claimed

Date of the actual completion of the international search

01 MAY 2000

Date of mailing of the international search report

02 JUN 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

KIMBINH NGUYEN

Telephone No. (703) 305-9683

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/02786

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

WEST

search terms: 3D model surface, geometric surface, pixels, mesh, displacement, filter, bind or combine, multi-memory, pipeline, rendering